

WHAT IS CLAIMED IS:

1. An apparatus for converting data of a first data format used on a particular computer into data of a second data format which is common to a plurality of computers in a distributed computing environment, comprising:

a caching part for storing a correspondence between data of the first data format and corresponding data of the second data format;

a marshalling part for discriminating whether at least a part of the data of the first data format to be converted has been stored in said caching part or not, for converting said data of the first data format to be converted into the data of the second format by using the data on said caching part when said data exists, and for converting said data to be converted into the data of the second data format without using said caching part when said data does not exist; and

a cache registering part for registering said converted data into said caching part.

2. An apparatus for converting data of a second data format which is common to a plurality of computers into data of a second data format used on a particular computer in a distributed computing environment, comprising:

a caching part for storing a correspondence between the data of the second data format and the data of the first data format;

an unmarshalling part for discriminating whether at least a part of the data of the second data format to be converted has been stored in said caching part or not, for converting said data of the second data format to be converted into the data of the first data format by using the data on said caching part when said data exists, and for converting said data of the second data format to be converted into the data of the first data format without using said caching part when said data does not exist; and

a cache registering part for registering said converted data of the first data format into said caching part.

3. A method of converting client program data of a client program into request communication data common to a plurality of program languages or computers in order to request a processing task from said client program to a server program on a server in a distributed computing environment, comprising the steps of:

discriminating whether at least a part of said client program data exists on a cache of said client program or not by referring to said cache;

converting said client program data into said request communication data by using the program data existing on said cache when said data exists and converting said data based on attributes of said client program data when said data does not exist;

storing the request communication data into

said cache as a conversion result when said data does not exist; and

transmitting said converted requesting communication data to said server program.

4. A method of converting request communication data which is transmitted from a client and is common to a plurality of program languages or computers into program data of a program language describing a server program which operates on a server in a distributed computing environment and processing a processing task requested by the client, comprising the steps of:

receiving said request communication data by an unmarshalling part in said server;

discriminating whether at least a part of said request communication data exists in a cache of said server program or not by referring to said cache;

converting said received requesting communication data into the program data of the program language describing said server program by using the request communication data existing in said cache when said data exists in said cache and converting said data based on attributes of said received requesting communication data when said data does not exist;

storing the program data into said cache as a conversion result when said data does not exist; and

processing the processing task requested by said client on the basis of said converted program data.

5. A method according to claim 4, wherein said

step of processing the processing task requested by said client on the basis of said converted program data further comprising the steps of:

processing the processing task requested by said client on the basis of said converted program data and forming the resultant program data;

discriminating whether at least a part of said resultant program data exists in said cache or not by referring to the cache in said server program;

converting said resultant program data by using the program data on said cache into response communication data which is common to the plurality of computers or program language when said data exists, while converting said data by determining attributes of said resultant program data when said data does not exist;

storing the response communication data into said cache as a conversion result when said data does not exist; and

transmitting said converted response communication data to said client program.

6. A method whereby a client program receives and processes response communication data which is transmitted from a server and is common to a plurality of computers or program languages in a distributed computing environment, comprising the steps of:

receiving said response communication data by said client program;

discriminating whether at least a part of said

response communication data exists on a cache of said client program or not by referring to said cache;

converting said received response communication data into client program data by using the communication data on said cache when said data exists and converting said data by discriminating attributes of said received response communication data when said data does not exist; and

storing the client program data into said cache as a conversion result when said data does not exist.

7. A computer software product including a computer-readable medium having a computer readable program embodied in the medium for making program data to be transmitted subject to a marshalling process to generate request communication data, said medium having stored thereon:

a program code portion for comparing said program data to be transmitted with said program data on a cache, said cache storing a pair of contents of said program data and contents of the communication data corresponding thereto for every type of the program data;

a program code portion for, when the contents of said program data to be transmitted and the contents of the program data on said cache are matched, comparing subsequent data to be transmitted with said program data on said cache until a difference is detected;

a program code portion for copying the communication data on said cache corresponding to the program

data which is matched until said difference is detected into requesting communication data;

a program code portion for, when a difference is detected between contents of certain program data and contents of the program data on the cache, forming corresponding communication data from the program data in which said difference is detected in accordance with a type of the program data in which said difference is detected;

a program code portion for storing a correspondence between the program code in which said difference is detected and said formed communication data onto said cache; and

a program code portion for, when there is no transmission program data to be compared, copying the data remaining on said cache into said request communication data.

8. A product according to claim 7, wherein said program code portion for forming the corresponding communication data from said program data in which the difference is detected further has:

a program code portion for discriminating whether the type of said program data in which the difference is detected is a basic type or a user definition type; and

a program code portion for, when the type of said program data in which the difference is detected is said basic type, recognizing an architecture or an

identifier of said program data in which the difference is detected and converting into the communication data.

9. A product according to claim 7, wherein said program code portion for forming the corresponding communication data from said program data in which the difference is detected further has:

a program code portion for discriminating whether the type of said program data in which the difference is detected is a basic type or a user definition type; and

a program code portion for, when the type of said program data in which the difference is detected is said user definition type, recursively calling said marshalling process by using said program data in which the difference is detected as a parameter.

10. A computer software product including a computer-readable medium having a computer readable program embodied in said medium for making received request communication data subject to an unmarshalling process to generate output program data, said medium having stored thereon:

a program code portion for comparing said received request communication data with said communication data on a cache in units of a predetermined communication data length, said cache storing a pair of contents of said program data and contents of the communication data corresponding thereto for every type of the program data;

a program code portion for, when the contents of said received request communication data and the contents of the communication data on said cache are identical, comparing subsequent received request communication data with said communication data on said cache until a difference is detected;

a program code portion for copying the communication data on said cache corresponding to the received request communication data which is matched until said difference is detected into said output program data;

a program code portion for, when a difference is detected between contents of certain received request communication data and contents of the communication data on the cache, forming corresponding program data from said received request communication data in which said difference is detected in accordance with a type of said received request communication data;

a program code portion for storing a correspondence between said received request communication data in which said difference is detected and said formed program data onto said cache; and

a program code portion for, when there is no received request communication data to be compared, copying the data remaining on said cache into said output program data.

11. A product according to claim 10, wherein said program code portion for forming the corresponding output program data in accordance with the type of said received



request communication data in which the difference is detected further has:

a program code portion for discriminating whether the type of said received request communication data in which the difference is detected is a basic type or a user definition type; and

a program code portion for, when the type of said received request communication data in which the difference is detected is said basic type, converting said data into the output program data by discriminating attributes of said received request communication data in which the difference was detected.

12. A product according to claim 10, wherein said program code portion for forming the corresponding output program data in accordance with the type of said received request communication data in which the difference is detected further has:

a program code portion for discriminating whether the type of said received request communication data in which the difference is detected is a basic type or a user definition type; and

a program code portion for, when the type of said received request communication data in which the difference is detected is said user definition type, recursively calling said unmarshalling process to said received requesting communication data in which the difference is detected.

13. An apparatus according to claim 1, wherein said

first data format is a data format specific to a particular program language and said second data format is a stream data format.

14. An apparatus according to claim 2, wherein said first data format is a data format specific to a particular program language and said second data format is a stream data format.

15. A method according to claim 3, wherein said client program data is specific to a particular program language describing said client program and said common communication data is stream data.

16. A method according to claim 4, wherein said program data of a program language describing a server program is specific to a particular program language describing a server program.

17. A method according to claim 3, wherein said method is stored on a computer-readable medium as a computer readable program.

18. A method according to claim 4, wherein said method is stored on a computer-readable medium as a computer readable program.